Re-use of an ontology for modelling urban energy systems

Koen H. van Dam, *Member, IEEE,* and James Keirstead

***Abstract*— The use of ontologies for the interoperability of software models is widespread, with many applications also in the energy domain. By formulating a shared data structure and a definition of concepts and their properties, a language is created that can be used between modellers and—formalised in an ontology—between model components. When modelling energy systems, connections between different infrastructures are critical, e.g. the interaction between the gas and electricity markets or the need for various infrastructures including power, heat, water and transport in cities. While a commonly shared ontology of energy systems would be highly desirable, the fact is that different existing models or applications already use dedicated ontologies, and have been demonstrated to work well using them. To benefit from linking data sources and connecting models developed with different ontologies, a translation between concepts can be made. In this paper a model of an urban energy system built upon one ontology is initialised using energy transformation technologies defined in another ontology, thus illustrating how this common perspective might benefit researchers in the energy domain.**

I. INTRODUCTION

Ontologies, i.e. formalised conceptualisations [8], are a proven tool for interdisciplinary modelling, providing an indispensable shared formal language. They facilitate consistent software design and interoperability between models, even when they have been built by different modellers, working with different techniques and in different domains. In energy systems multiple infrastructures are interlinked and modellers could therefore benefit from the interface provided by a shared ontology in order to access disparate data sources and connect models.

The energy modelling community has begun to recognize this need and some initial work has been done. Keirstead and van Dam [15] concluded that "we would be interested in establishing an open community effort to build a standardised modelling ontology" for energy systems, based on the lessons learnt from a demonstration that an energy conversion technology from one ontology could successfully be described in the other even though the exact properties and classes used were different. Such a standard and shared ontology could have great benefits to the modelling community as it would enable stronger cooperation between groups and disciplines. Similarly Catterson *et al.* [2] advocate the development of a shared ontology for power systems, saying "[...] the community of researchers working in this area must agree on the following points: Standards for data exchange [...] and [...] creation of an upper ontology for smart grid terms and concepts, likely based on existing data standards [...]", referring to the work of the IEEE Power Energy Society trying to address these issues [16]. So far this work as not been open—a main requirement for widespread use and community involvement—but there are plans to release the standards shortly.

There are however two pre-requisites for the development of a shared ontology: that researchers have a shared view of the world, and that they have comparable aims for which they want to use the ontology. In the ontology definition phase, the first fundamental rule is that "there is no one correct way to model a domain—there are always viable alternatives" [19]. Guarino [9] stresses the *intended meaning* in his definition of an ontology. If an ontology is to be used by people with a different world view (e.g. a different valuation on what is important and what does not need to be emphasized) then the resulting ontology may end up being only generic and without much expressive power for any application. When these two conditions (shared view of the world and shared aim) have been met, a joint effort to developing an ontology may be fruitful.

Although it is our aim to design a high-level ontology for modelling energy systems, it has to be acknowledged that researchers already have existing tools and models which incorporate ontologies that may be closely related, but not the same. The challenge therefore is to develop an interface which allows the re-use of elements from one ontology in another, while providing a uniform representation of the domain. The goal of this paper is to begin this process by identifying the major concepts that might be part of such an ontology and to explore some of the associated practical issues.

The paper is structured as follows. First in Section II the background of two existing ontologies considered in this paper is briefly sketched, after which we discuss how ontologies can be connected and how interoperability can be provided (Section III). Using a case study presented in Section IV, we demonstrate how the the use of a master ontology enables elements from both ontologies to be used in a single model of an urban energy system. Section V concludes with a discussion on the usefulness of the approach and we discuss how ontology couplings can be done more easily in the future, particularly drawing attention to a new initiative to develop a high-level energy systems modelling ontology.

K.H. van Dam is with the Faculty of Technology, Policy and Management of the Delft University of Technology, 2600 GA Delft, The Netherlands k.h.vandam@tudelft.nl

J. Keirstead is with the Department of Chemical Engineering of Imperial College, South Kensington Campus, London, SW7 2AZ, UK j.keirstead@imperial.ac.uk. He would like to acknowledge the support of BP through the BP Urban Energy Systems project at Imperial College London.

978-1-4244-8479-9/10/$26.00

II. BACKGROUND

Before going further, we should clarify what we mean by an "energy systems model". By "energy system" we have in mind Jaccard's [12] broad definition: "the combined processes of acquiring and using energy in a given society or economy" (p. 6). Models of course mean different things to different people, but in this context we are primarily concerned with quantitative models for the analysis, prediction, exploration and study of different scenarios to support decision making. This field therefore encompasses a wide range of disciplines and modelling tools; some practitioners may be using spreadsheets to examine aggregate national consumption statistics, while others may use detailed simulation software to assess the performance characteristics of a specific energy technology. Yet all of these applications imply an underlying conceptualization of the major elements of an energy system and their associated attributes; ontologies provide the tools to make these descriptions explicit.

In this paper two different energy ontologies, developed independently from each other, are used. They are briefly introduced below.

A. UES ontology

SynCity (short for "Synthetic City") is a modelling system for urban energy systems developed at Imperial College London [13]. The goal of SynCity is to provide a platform for the modelling of urban energy systems (UES) at multiple scales. This requires the use of several different modelling techniques, including mathematical programming and agent-based modelling. Within this context, the UES ontology was introduced to provide consistent class definitions between the models and for the storage and management of system components. The UES ontology consists of a number of object classes that describe the main elements of an urban energy system, as well as specific instances of these classes. Within the context of this paper, two classes are highlighted:

Resources such as electricity or natural gas, are described by a series of physical, economic, and model attributes. These include mass and energy densities, unit prices, or maximum stock values.

Processes are technologies that convert one set of resources into another set. There are multiple subclasses to describe simple conversion technologies as well as more complex transportation and storage processes.

The ontology also contains detailed classes for the definition of the physical infrastructure of a city and it features a number of supporting classes, including the `Unit` class and its instances, designed with the JScience library [3] in mind to facilitate easy unit conversion.

B. STS Ontology

To support the development of agent-based models of socio-technical systems (STS), the STS ontology has been developed at the Delft University of Technology [21]. The aim was to build an ontology not for one specific application domain (e.g. an electricity infrastructure), but to find commonalities between applications and therefore to develop a modelling framework that is able to deal with the reality of socio-technical network systems that are interconnected across sectors. Modellers use the ontology to formalize domain knowledge, as language in the definition of behavioural rules and as communication protocols between agents. In this way, parts of the model can be re-used (e.g. re-using the model of a certain technology with a different agent, or re-using behavioural rules of one agent in another one, or even copying complete agents into another model) and models of different infrastructures can be connected, even when they are developed by different modellers. For the purpose of this paper, the main classes are the following:

Technologies follow the input-output paradigm to define energy or mass conversions. It can use different recipes consisting of different input-output pairs to reflect different modes of operation. Properties define, for example, the capacity of the technology to produce a certain product, the maintenance and operational costs attached to its operation, and so on. Technologies are not active units but have to be operated by an *agent*, who makes decisions about how to use the technology.

GoodNames describe the "products" that exist in the system (e.g. crude oil or electricity).

Additionaly, the ontology features a rich set of classes to describe agents, different types of contracts, and the physical infrastructure as well as the actual flows in the system.

C. Comparison

Each ontology was designed with different initial goals in mind, but as demonstrated in [15], there are significants overlaps and compatible elements. The aim of the definition of the `Technology` and `Process` concepts is the same: to provide the inputs and outputs of (energy) conversion technologies with several properties, including costs, for use in models with which to assess different policies or configurations at the operational, tactical and strategic level. Both ontologies have been defined in Protégé Frames [7]. There are, however, also substantial differences and this creates difficulties when trying to use objects from one ontology within another modelling domain. While modellers know that `Technology` and `Process` are the same and that their properties, even though conceptualised slightly differently, are comparable, this does not mean the software applications "understand" this as well. To enable interoperability between the two ontologies and applications built using them, they need to be explicitly connected.

III. CONNECTING ONTOLOGIES

Haslhofer and Klas [11] present an extensive survey of techniques for obtaining interoperability between data formalised in different ways, highlighting the difference between *instance*, *schema* and *schema definition language* following the definition of meta levels in the Object Management Facility (OMF) specification [20]. We can map this to ontology instances, ontology definition, and ontology language, respectively. The ontology instances can be considered as level M0, the definition of the ontology as M1 and the

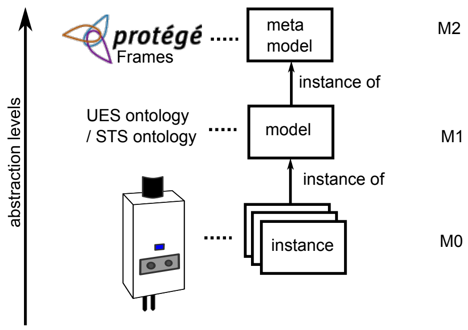
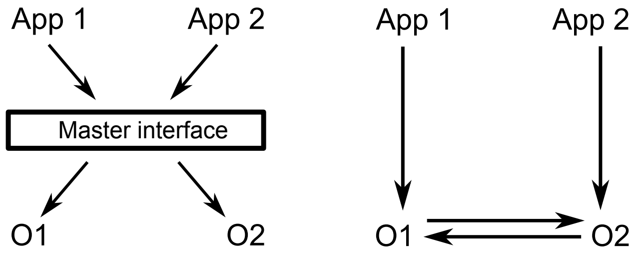


Fig. 1: Meta-levels in the definition of ontologies



(a) Ontology "master" strategy (b) Ontology "ad-hoc"

Fig. 2: A comparison of the "master" and "ad-hoc" ontology translation strategies

ontology language as M2 (See Figure 1). Interoperability is possible at these meta levels and depending on level of standardization, there are different mapping strategies: language, semantic and instance mapping [11]. Furthermore, Euzenat [5] distinguishes different levels of ontology interoperability: encoding, lexical, syntactic, semantic, and semiotic.

The two ontologies presented above are both based on the same ontology language (i.e. Protégé Frames) which means there is encoding, lexical and syntactic interoperability at the M2 level. To be able to use instances (M0 level) from both ontologies in one model, we need to develop a shared definition of the concepts used in the ontology and achieve interoperability at the M1 level (semantic and semiotic interoperability). Because the domain and intended meaning of the ontology definitions overlap (condition posed in [9]) and there is a consensus for integration (necessarily according to [10]), it should be possible to work towards a joint definition at the M1 level.

The work towards a joint ontology definition can be separated by level of generality into a top-level ontology (also called upper ontology), domain ontology and application ontology [9]. Some elements required in M1 for energy systems (e.g. the definition of units) are at the level of an upper ontology which go beyond the domain itself; see for example the SUMO ontology developed by an IEEE working group [17]. Other elements, such as the concepts for the definition of inputs and outputs for conversion technology, can be generalised at the domain level of energy systems. Finally, there are application specific concepts which do not relate to the domain but to implementation issues, for example properties related to the solver or optimizer.

So what approaches are available to build a common M1 from already existing ontologies? In [4] two major strategies for translating ontologies are presented. The first approach is to create a *master ontology* which encompasses the two ontologies being translated. However this approach requires that "a global ontology can cover all existing ontologies, and we can get agreement by all ontology experts to write translators between their own ontologies and this global ontology" [4](p. 5). Maintenance can therefore be difficult as new ontologies have to be reflected in the master ontology. The second technique is to translate between ontologies on an *ad-hoc basis*, directly converting from a source ontology to a target. This creates fewer problems with extensive abstraction and maintenance, but the complexity of the translation tasks increases exponential with the number of ontologies to be adapted.

The choice of an ontology translation strategy depends on the application context. If we choose to consider our two applications only, then the ad-hoc approach is preferable as it avoids the overhead of developing a master ontology. However, given the interest of the community in a wider framework and the close similarities between the two ontologies that each have proven to be useful in their own applications, we have decided to pursue a master ontology approach as a first step towards a standard ontology for modelling energy systems. The aim here is to create an object-oriented programming solution for translating between energy system ontologies, using the master ontology concept.

Figure 2 illustrates the advantages of such an approach. In Figure 2a, a number of object-oriented applications (App_1, App_2) each connect to the master ontology interface. This layer then connects with specific ontology implementations (O_1, O_2), allowing applications to easily switch between ontologies and access data contained within their respective knowledge bases. Figure 2b shows the ad-hoc approach. Since each application is designed to use its own ontology only, other applications must use this specific format and write customized translation services between each of the other ontologies.

IV. CASE STUDY

To illustrate these concepts, we present a simple case study in which two ontologies are used in a single model of an urban energy system.

A. Set-up

Using the SynCity tool kit, we wish to design a gas and electricity distribution system for a small city (see for example [14]). The end use demands for heat and power are specified as input to the model, along with the properties of key objects necessary to complete the supply system: the unit cost of electricity and gas distribution networks; the costs of these resources; and the cost and performance of a conversion technology, namely a 25 kW domestic gas boiler. The technology as well as the resources are described in an ontology knowledge base. This input data is then used by

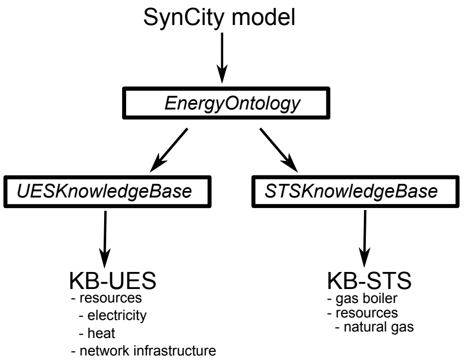


Fig. 3: Set-up of the case study with the EnergyInterface

a mixed-integer linear programming model to determine the lowest cost energy supply system, which is described by the routing of the resource distribution networks. The model also determines the number, locations and operating rates of the gas boilers in this scenario.

As described above, the SynCity tool kit is supported by the UES ontology. Its knowledge base contains a boiler, which is considered as an instance of the `ConversionProcessType` class. However, in this case study instead of reading this object from the UES ontology, we instead read the boiler's attributes from an instance of a `Technology` class of the STS ontology (See Figure 3). Some of the resources, including electricity and heat, are read from the UES ontology while the resources used in the definition of the boiler, including natural gas, were read from the STS ontology.

The aim of the case study is to replicate the results of a set-up which only uses the UES knowledge base with the set-up from Figure 3. If the results are the same, we can conclude that the interface and the master ontology are working correctly.

B. The `EnergyOntology` interface

In practice, implementing the master ontology strategy means creating an *interface* in Java. An interface declares the expected behaviour of an object type in Java, but does not provide the implementation of specific methods. Modellers would therefore develop their applications using this master interface and leave the developers of each sub-ontology to provide the details of the translation process. An advantage of this approach is that each implementation can extend the interface as needed to define model specific additional features.

C. Ontology translation

The manual translation process proceeded as follows:

1) An `EnergyOntology` interface was created in Java. This declares two methods, `getResource` and `getTechnology`, which return the appropriate objects from an ontology given a key string for the label, e.g. "boiler".
2) Two classes implementing this interface were created: `UESOntology` and `STSOntology`. In `UESOntology`, the existing SynCity methods for loading resources and processes from the UES knowledge base were specified. In `STSOntology`, additional code had be written to explicitly state which STS ontology classes correspond to `Resource` and `ConversionProcessType` in the UES knowledge base. In this implementation, each of the input fields are manually processed to correspond to the SynCity Java objects. As an example of such a translation, consider the following: the UES knowledge base assumes that properties have a String field called "name" which defines their purpose (e.g. energy density, capital cost, etc.). In the STS knowledge base, this data is implied by the class of the property, e.g. `CapitalCost` or `EnergyDensity`.
3) The source code of the model was modified to use the generic `EnergyOntology` master interface. To load the boiler technology, `STSOntology` was used; to load other data, `UESOntology` was used. This code excerpt is shown below.

```
// Get gas boiler and related resources from STS ontology
EnergyOntology sts = new STSOntology();
Technology boiler = sts.getTechnology("boiler");

// Get electricity resource from UES ontology
EnergyOntology ues = new UESOntology();
Resource elec = ues.getResource("elec");
```

D. Results

After the design for the master interface was implemented, the model could be initialised using instances from both ontologies. The model was run to create the resource and technology network for the city as described in [14]. The resulting network and model metrics were compared with the original implementation in SynCity which uses only the UES ontology. The model outcomes were the same both cases, confirming that the interface works.

E. Issues raised

Although this solution worked, it is by no means elegant. To illustrate the general perils associated with this sort of integration exercise, a concrete example is provided which highlights two problems, namely the semantic equivalence of objects and class extensions.

To set the scene, when the code listed above is run the `sts` object loads the gas boiler from the STS ontology into the model. This boiler object is defined in part by properties related to its resource inputs and outputs and accordingly, the resources for natural gas, output heat, and waste heat are also loaded from the STS knowledge base. Subsequent code within the SynCity model then attempts to modify these same resources.

The first problem is object equivalence (i.e. *semantic equivalence*). In many modelling applications, including SynCity, it is important that a unique definition of certain objects is maintained in order to facilitate consistency and to allow interaction between model components. In our model, where natural gas is transported through a distribution network and used in a domestic boiler, the 'gas' resource should be the same for both the network and conversion technology. In Java, this behaviour can be ensured by specifying the definition of object equivalence with customized `equals` and `hashCode` methods.

The question therefore is how to define this equivalence. Typically two objects would be compared on a field-by-field basis so that, for example, if two resources have the same name, unit cost and so on, then they are equivalent. Yet in our case, the STS and UES 'gas' objects have different names and hence are seen by Java as different objects. A naïve solution is to rename one of the instances to match the other. However this is clearly undesirable as it may cause problems for native applications within each system that rely on these labels. Furthermore, suppose that all fields were 'substantially' equivalent (e.g. energy density of the two resources was only 0.1% different): is this difference significant enough to warrant a comparison failure? Therefore a key element of developing a master interface and ontology is to define the semantic equivalence of objects.

The second problem is *class extension*. In the translation literature cited above [4], extension refers to the ability to derive subclasses of a second ontology. That is, given ontologies O_1 (with a sub-ontology O_{1_s}) and O_2, how can we derive O_{2_s}? The issue here, however, is that not all of the attributes in one ontology might be present in the other, and again there may be restrictions both structural and functional that prevent us from adding these new features to our secondary ontology. For example, the UES ontology includes several fields which are specific to the implementation of the optimization model (i.e. they are part of the application ontology rather than the domain ontology). These fields have no equivalent with the STS ontology.

This issue highlights the difference between application and domain ontologies and suggests the appropriate class extension strategies for each. Again, our interest here is at the domain ontology level. Therefore if the domain ontology (i.e. as encapsulated by the `EnergyOntology` interface) specifies that all technologies should have a property called "footprint", then the Java interface technique will ensure that developers explicitly provide a value for this property. In other words, if the knowledge base complying with the master interface lacks this field, then some default value must be explicitly assumed within the implementing subclass (e.g. `STSOntology`). Any additional data required at the level of the application (e.g. the optimization parameters for SynCity) *may* be found in a particular knowledge base that complies with the domain ontology but the developer cannot rely upon this behaviour and may again need to assume default values or provide another solution to deal with these missing parameters.

V. DISCUSSION

This brief example sheds light on some of the advantages and disadvantages of merging ontologies. On the one hand, it is possible to incorporate data from another application enabling the re-use of data, thus saving the modeller significant time and broadening the scope of modelling activities. This is particularly useful when modelling urban systems, since studying the diverse issues in cities requires various smaller models rather than building one super model [1]. Furthermore, there is a strong need to incorporate different kinds of data, such as environmental (e.g. emission data), socio-economic (e.g. profiles of households) and technical (e.g. descriptions of energy technologies) which are being compiled by different institutes at the local, national, and international level in different languages and formats.

A few general issues can be addressed. First of all, the automated or half-automated merging and translation of ontologies is a difficult subject and, while a lot of work is done (e.g. [18], [6]), the literature seems to agree that extensive manual curation is still required to fine-tune different conceptualisations and this depends on a possibly slow consensus building process. In light of this, and given that ontologies in the energy domain are becoming more popular but there are at present no widely used standards, now is a good time to jointly work on standardization. We therefore very much welcome the ongoing efforts in this direction (e.g. [16]).

It should be stressed that the aim of models developed by power engineers may be very different from the goals of modellers comparing higher level systems such as infrastructure interactions and urban energy systems and a different level of detail may be required. For example, to model the various energy flows within a city it might not be required to have detailed concepts to describe a circuit breaker in the electricity network. Even though the application domain is the same or highly related, a different aim could result in incompatibility of the conceptualisation. Still, we would hope there are sufficient connections and overlap that part of the conceptualisation (e.g. at the top-level of an engineering ontology with units, materials, resources) could be shared.

Nevertheless, given the interest in the energy modelling community and the insights gained from this brief experiment, we believe that it is worthwhile to pursue a community-effort to facilitate the interoperability of energy modelling knowledge bases by developing a shared domain ontology (i.e. for energy systems), based on engineering upper-ontologies (e.g. units). There are number of follow-up steps that could be taken:

1) Conduct a survey within the energy modelling community to find out what standards and ontologies are being used and for which modelling domains.
2) Identify the common areas of practice. One particular starting point is the use of a standard framework for measurement units. Again, this is an area where other organizations are beginning to draw together standards so it makes sense to take advantage of this. This may also mean restricting the scope of the modelling system to particular application domains, e.g. strategic energy systems design versus more detailed operational micro-simulation.
3) Create a working arrangement for the design and maintenance of the high-level framework. This could be modelled on existing standards bodies such as the

IEEE or W3C. Key areas to be resolved include how to extend the master framework and a series of good practice suggestions for model developers.

It should be stressed that we are calling for cooperation between researchers towards the development of a standard. While the two ontologies briefly introduced in this paper have proven useful in their own niches, their greatest value is unlikely to be as a template for a master ontology but rather as a case study of the costs and benefits of ontology integration. Finally, such a shared ontology does not necessarily have to be universally accepted, but there is already a clear benefit within a smaller community.

REFERENCES

[1] M. Batty. Extending the focus of urban modelling tools, 2010.

[2] V. M. Catterson, E. M. Davidson, and S. D. J. McArthur. Agents for Active Network Management and Condition Monitoring in the Smart Grid. In *Proceedings of the First International Workshop on Agent Technologies for Energy Systems (ATES 2010)*, pages 13–18, 2010.

[3] J.-M. Dautelle. JScience library. `http://jscience.org`, 2009.

[4] D. Dou, D. McDermott, and P. Qi. *Ontology translation on the semantic web*, page 3557. Springer, 2005.

[5] J. Euzenat. Towards a principled approach to semantic interoperability. In *Proceedings of the IJCAI-01, Workshop on Ontologies and Information Sharing, Seattle, USA*. Citeseer, 2001.

[6] J. Euzenat. An API for Ontology Alignment. *The Semantic Web ISWC 2004*, pages 698–712, 2004.

[7] J. H. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubezy, H. Eriksson, N. F. Noy, and S. W. Tu. The evolution of Protégé: an environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, 58(1):89–123, January 2003.

[8] T. Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5:199, 1993.

[9] N. Guarino. *Formal ontology and information systems*, chapter 1. IOS Press, 1998.

[10] B. Haslhofer. A Comparative Study of Mapping Solutions for Enabling Metadata Interoperability. `http://www.cs.univie.ac.at/upload/550/papers/haslhofer08\_mapping\_solutions\_TR.pdf`, 2008.

[11] B. Haslhofer and W. Klas. A survey of techniques for achieving metadata interoperability. *ACM Computing Surveys (CSUR)*, 42(2), 2010.

[12] M. K. Jaccard. *Sustainable Fossil Fuels: The Unusual Suspect in the Quest for Clean and Enduring Energy*. Cambridge University Press, Cambridge, 2005.

[13] J. Keirstead, N. Samsatli, A. M. Pantaleo, and N. Shah. Evaluating integrated urban biomass strategies for a UK eco-town. In *Proceedings of the European Biomass Conference*, pages 2115–2127, Hamburg, 2009.

[14] J. Keirstead, N. Samsatli, N. Shah, C. Weber, and D. Fisk. The Implications of CHP Planning Restrictions on the Efficiency of Urban Energy Systems. In *23rd International Conference on Efficiency, Cost, Optimization, Simulation and Environmental Impact of Energy Systems*, Lausanne, 2010.

[15] J. Keirstead and K. H. van Dam. A comparison of two ontologies for agent-based modelling of energy systems. In *Proceedings of the First International Workshop on Agent Technologies for Energy Systems (ATES 2010)*, pages 21–28, 2010.

[16] S. D. J. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi. Multi-Agent Systems for Power Engineering Applications—Part 2: Technologies, Standards and Tools for Building Multi-Agent Systems. *IEEE Trans. Power Systems*, 22(4):1753–1759, November 2007.

[17] I. Niles and A. Pease. Towards a standard upper ontology. *Formal Ontology in Information Systems*, 2001.

[18] N. Noy and M. Musen. The PROMPT suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024, 2003.

[19] N. F. Noy and D. L. McGuinness. Ontology Development 101: a Guide to Creating Your First Ontology. Technical Report SMI-2001-0880, SMI, 2001.

[20] OMG. Meta Object Facility (MOF) Core Specification 2.0. `http://www.omg.org/spec/MOF/2.0/`, 2006.

[21] K. H. van Dam. *Capturing socio-technical systems with agent-based modelling*. PhD thesis, Delft University of Technology, Delft, the Netherlands, 2009.